TOOL 64

- ★ Mycket kraftfullt programmerings- och buggsökningshjälpmedel
- ★ Nyckelord som underlättar skärmhantering
- ★ DOS-supportprogram



S

for

COMMODORE 64

INNEHÅLLSFÖRTECKNING

1. ÖVERSIKT

TOOL 64 är ett kraftfullt komplement till din COMMODORE 64. Som både underlättar och förbättrar ditt programmeringsarbete.

Den innehåller en massa nya kommandon av BASIC–typ som kommer att förenkla hanteringen av ditt datorsystem. Till exempel att rita linjer, att rita högupplösningsgrafik och använda tecken på skärmen på en gång, att hantera skärmsidor på diskettstationen.

TOOL 64 innehåller även hjälpinstruktioner för att göra och felsöka program, och att hantera strängar. Det finns även en funktion för joysticks.

DOS har lagts in för att underlätta dit arbete med diskstationen.

En noggrann läsning av denna manual kommer att göra det möjligt för dig att effektivt använda TOOL 64. Det underlättar om du samtidigt går igenom exemplen på din COMMODORE 64.

INSTALLATION

Kontrollera att strömmen är avslagen. TOOL 64 är monterad i en ROM-kassett som man ansluter till minnesexpansionsporten på baksidan av sin COMMODORE 64. Vänd kassetten så att ettiketten kommer uppåt. När du slår på strömmen så laddas TOOL 64 automatiskt in i minnet. Du har nu 30k arbetsminne kvar och kan börja arbeta med din TOOL 64.

2. BILDSKÄRMSGENERATORN

2–1 Översikt

Bildskärmsgeneratorn som finns i TOOL 64 är konstruerad för att underlätta datainmatningen på bildskärmen. Inmatningen blir pålitligare än vanliga BASIC-rutiner och utförandet, huvudsakligen i skärmhanteringen, mycket snabbare eftersom den är gjord på maskinspråk.

TOOL 64 kontrollerar följande funktioner :

– Skärmkommandon att rita linjer horisontella och vertikala, rulla skärmen eller del av skärmen, etc.

– Inmatningskommandon att definiera inmatningszoner, att överföra innehållet i zoner till BASIC-variabler, etc.

– Hantering av skärmsidor att spara och ladda skärmsidor från diskett.

Skärmen är definierad som ett ritbord med origo i det övre vänstra hörnet. Koordinaterna för origo är 1,1. En punkt på skärmen definieras av: l =radnummer ; c =kolumnnummer. Skärmen består av 25 rader och 40 kolumner.

En skärmsida är allt som finns på skärmen i ett givet ögonblick. Det är möjligt att skapa en skärmsida och spara den på disk. Sedan kan man ladda in den direkt på skärmen och komplettera den om det behövs. Det är också möjligt att använda både tecken och grafik samtidigt.

Skärmkommandon

2–2 tline

FUNKTION: tline ritar en horisontell linje på skärmen vars utseende bestäms av startpunkten och längden.

UPPBYGGNAD: tline ln, l, c
där ln =längden på linjen
l =radnummer för startpunkten
c =kolumnnummer för startpunkten

ANVÄNDNING: kan användas för att rita ramar på skärmen eller att stryka under text

EXEMPEL: tline 22,1,1
tline 22,12,1
Dessa två kommandon ritar två parallella linjer i övre delen av skärmen

SKÄRMKOMMANDON

2–3 tcol

FUNKTION: tcol ritar vertikala linjer på skärmen vars utseende bestäms av startpunkten och längden

UPPBYGGNAD: tcol ln, l, c
där ln =längden på linjen
l =radnummer för startpunkten
c =kolumnnummer för startpunkten

ANVÄNDNING: kan användas för att rita ramar (tillsammans med tline).

EXEMPEL: tcol 12,1,1
tcol 12,1,22
dessa två kommandon ritar två parallella linjer i övre delen av skärmen
10 tline22,1,1:tline22,12,1
20 tcoll2,1,1,1:tcoll2,1,22
dessa två kommandon ritar upp en ram i skärmen

SKÄRMKOMMANDON

2–4 clear

FUNKTION: clear raderar skärmen med början från en startpunkt.

UPPBYGGNAD: clear l, c, lg, ln
där l =radnummer för startpunkten
c =kolumnnummer för startpunkten
lg =höjden (antalet linjer)
ln =längden (antalet kolumner)

ANVÄNDNING: kan användas för att radera delar av skärmen.

EXEMPEL: clear 1,1,1,40
raderar första raden på skärmen
30 clear 1,1,12,22
tar bort ramen som vi ritade i förra exemplet (även det som finns innanför). Raderar alltså ett fönster som är 12 rader och 22 kolumner stort.

SKÄRMKOMMANDON

2–5 out

FUNKTION: out visar en sträng(a$) på en bestämd plats på skärmen

UPPBYGGNAD: out a$, l, c
där a$ =någon sträng
l =radnummer för startpunkten
c =kolumnnummer för startpunkten

ANVÄNDNING: kan användas för att visa en sträng varsomhelst på skärmen utan att man behöver använda "CRSR"-tangenterna i strängen. Vilket underlättar om man vill fylla i ett fönster.

EXEMPEL: 40 a$="ANDERS ANDERSSON"
50 out a$,5,10
dessa två rader kommer att visa ANDERS ANDERSSON på 5:e raden med början i 10:e kolumnen.

SKÄRMKOMMANDON

2–6 rev

UPPBYGGNAD: REV l,c,lg,ln,f
där l = radnummer för startpunkten
c = kolumnnummer för startpunkten
lg = höjden (antalet linjer)
ln = längden (antalet kolumner)
f = färg (0–15)

FUNKTION: ändrar bakgrundsfärgen i ett fönster bestämt av startpunkt, längd, höjd och färg.

ANVÄNDNING: kan användas för att dra uppmärksamhet till ett visst område på skärmen. Genom att använda kommandot flera gånger med samma koordinater, kan man få ett fönster att blinka. Blinkhastigheten kan regleras med en for:next loop.

EXEMPEL:

```
140 rev5,10,1,16,3
```

ändrar bakgrundsfärgen på namnet i förra exemplet.

SKÄRMKOMMANDON

2–7 scroll

FUNKTION: scroll flyttar ett fönster uppåt, nedåt, till höger eller vänster.

UPPBYGGNAD: scroll l, c, lg, ln, ty
där l =radnummer för startpunkten
c =kolumnnummer för startpunkten
lg =höjden (antalet rader)
ln =längden (antalet kolumner)
ty =typ av scrollning u =upp
d =ner
l =vänster
r =höger

ANVÄNDNING: kan användas för att scrolla fram en stor mängd data i ett fönster.

EXEMPEL:

```
10 clear1,1,25,40
20 a$="COMMODORE 64 "
30 for i=1to13
40 scroll 25,3,1,13,r
50 scroll 13,3,13,1,d
60 scroll 1,3,1,13,r
70 scroll 1,3,13,1,u
80 scroll 13,15,13,1,d
90 scroll 13,3,1,13,r
```

```
100 out mid$(a$,i,1),13,3
110 scroll 13,38,13,1,d
120 scroll 1,38,13,1,u
130 scroll 1,26,13,1,u
140 scroll 13,26,1,13,l
150 out mid$(a$,i,1),13,38
160 for j=1to222:next j
170 next i
180 goto30
```

INMATNINGSKOMMANDON

2–8 Översikt

Inmatningskommandona i TOOL 64 gjordes för att ersätta INPUT och GET-satserna i vanlig BASIC. Dom underlättar för dig att göra effektiva och pålitliga inmatningsrutiner. Den här delen kommer att behandla dessa kommandon. Man definierar en zon med vissa parametrar (decz).

När den är definierad så talar man om för TOOL 64 att den ska hämta data till zonen (reqz). Sedan överför man datat till en sträng (inz).

Eller annars kan man också överföra data från en sträng till en zon (outz). Och slutligen kan man radera datat i en zon (clearz).

INMATNINGSKOMMANDON

2–9 decz

FUNKTION: decz används för att definiera alla parametrar för en zon med kontrollerad inmatning.

UPPBYGGNAD: DECZ n,l,c,ln,ty,f$

där n = identifikationsnummer (0<=n<=255)
l = radnummer för startpunkt (1<=1<=40)
c = kolumnnummer för startpunkt (1<=lc<=23)
ln = zon längd (1<=ln<=255)
ty = typ av zon
där n = numeriska data
m = data med versaler
r = andra möjligheter att lämna zonen förutom RETURN (för en numerisk zon, nästan alla tangenter förutom de numeriska; för en alfanumerisk zon, alla icke alfanumeriska tangenter)
p = zon definerad av PRINTUSING (f$) (utgår om f$ icke önskas).
f$ = format av numerisk zone e.g. "1999999.99"

ANVÄNDNING: decz gör det möjligt att enkelt och exakt bestämma vad som skall matas in i en zon

EXEMPEL: DECZ 2,1,2,9,n,p,"199999.99"
Detta exempel definierar en zon (nummer 2) med startpunkt (1,2) och längd (10 tecken). Endast siffror accepteras (n).
Printusingformat används (p), tecken skrivs endast ut om det är minus (1) och det kommer alltid att finnas två tecken efter decimalkommat (88), nollor om inget annat skrivs in. Talet består av 5 siffror före, (99999), och två efter decimalpunkten.

ANTECKNINGAR:
Mer än en sort kan kombineras i samma decz, följande möjligheter accepteras

n	:endast siffror	–RETURN för att lämna zonen
n,r	:endast siffror	–En programmerbar tangent för att lämna zonen
n,p	:endast formaterade siffror	–RETURN för att lämna zonen
n,r,p	:endast formaterade siffror	–En programmerbar tangent för att lämna zonen
m	:endast versaler	–RETURN för att lämna zonen
m,r	:endast versaler	–En programmerbar tangent för att lämna zonen
r	:alla tecken accepteras	–En programmerbar tangent för att lämna zonen
p	:(samma som n,p)	
r,p	:(samma som n,r,p)	
	inget:alla tecken accepteras	–RETURN för att lämna zonen

INMATNINGSKOMMANDON

2–10 reqz
ANVÄNDNING

FUNKTION: reqz utför inhämtningen och använder sig de parametrar som definieras i motsvarande decz

UPPBYGGNAD: reqz n
där n =identifikationsnummer

ANVÄNDNING: reqz används tillsammans med decz.
Om motsvarande decz ej finns genereras ett felmeddelande. Med detta kommando tar TOOL 64 kontroll över datainmatningen enligt zondeklarationen. En cursor visas i början av zonen.

Funktionerna CLR, HOME, CRSR

VÄ och HÖ, INST/DEL kan användas. Om zontypen är definierad till r och koden (zo) för att lämna zonen skiljer sig från 13. Hoppar man sedan in i zonen igen kommer cursorn att återvända till utgångsläget

EXEMPEL: om det föregående exemplet finns kvar, slå in reqz 2 och se vad som händer. Endast siffror accepteras.

INMATNINGSKOMMANDON

2–11 inz

FUNKTION: inz överför innehållet i en zon till en sträng

UPPBYGGNAD: inz n,a$
där n =identifikationsnummer
a$ =vilken sträng som helst

ANVÄNDNING: inz används för att överföra alla data i en zon till en sträng när alla data är tillfredställande.

EXEMPEL: om de två tidigare exemplen finns kvar, mata in: inz 2,a$
nu är överföringen klar. För att kontrollera, mata in: print a$
Strängen till vilken man överför kommer att ha samma längd som zonen oavsett dess innehåll.

INMATNINGSKOMMANDON

2–12 outz

FUNKTION: outz visar en sträng i en zon

UPPBYGGNAD: outz n,a$
där n =identifikationsnummer
a$ =vilken sträng som helst

ANVÄNDNING: kan användas för att visa flera olika strängars innehåll i en zon.

EXEMPEL: om de tidigare exemplen finns kvar,
mata in: a$="11111"
outz 2,a$
reqz 2

ANTECKNINGAR: i det fall där en zon är definierad i printusing och det blir för många decimaler avrundas talet helt. Om det blir för många heltalssiffror så fylls zonen med "*****".

INMATNINGSKOMMANDON

2–13 clearz

FUNKTION: clearz raderar en zon

UPPBYGGNAD: clearz n1 (to n3)
där n1 och n3 =identifikationsnummer

ANVÄNDNING: clearz kan användas för att radera innehållet i en ruta utan att radera själva rutan samt efter en inmatning och före nästa inmatning

EXEMPEL: clearz 2

INMATNINGSKOMMANDON

2–14 carget

FUNKTION: carget hejdar programmet och väntar på att en tidigare definierad tangent ska tryckas ned.

UPPBYGGNAD: carget a$ (,l,c)
där a$ =sträng som innehåller tilllåtna tecken
l,c =position på skärmen där en cursor kommer att visas (behöver ej anges).

ANVÄNDNING: carget ersätter gammal BASIC-rad 100 get a$:if a$="" then 100 och tillför många fördelar.
100 carget "abc" hejdar programmet tills tangent a, b eller c trycks ned. Efter en carget antar variablerna ok och zo följande värden: ok visar position i a$ (a ger ok=1, c ger ok=3) zo visar ASCII-värdet (b ger zo=66, c ger zo=67)

EXEMPEL:

BASIC

```
100 get a$:if a$="" then 100
110 if a$="A" then gosub 1000:goto 200
120 if a$="B" then gosub 2000:goto 200
130 if a$="C" then gosub 3000:goto 200
140 if a$="Z" then gosub 4000:goto 200
150 goto 100
200 rem *** programmet fortsätter ***
```

TOOL 64

```
100 a$="ABCZ":carget a$,24,20
110 on ok then gosub 1000,2000,3000,4000
200 rem *** programmet fortsätter ***
```

ANTECKNINGAR: om a$="" accepteras alla tecken.
Om du använder l, c parametrarna kan du bestämma blinkhastigheten på cursorn. POKE 703,N där N = hastigheten (normal hastighet = 100)

KOMMANDON FÖR HANTERING AV SKÄRMSIDOR

2–15 Översikt

Många gånger kan strukturen på en skärm, ramar, rutor eller liknande vara användbara vid flera olika tillfällen. I vanlig BASIC måste man göra om skärmen varje gång som man ska använda den, vilket tar både tid och minnesutrymme i anspråk. Med TOOL 64 kan man spara skärmen och ta fram den vid behov, en hjälpskärm t ex. Man kan alltså byta ut en skärm man sitter och arbetar med. Man sparar den aktuella skärmen på disk och hämtar en hjälp-skärm t ex. När man är klar med hjälp-skärmen hämtar man upp den aktuella skärmen igen och cursorn kommer att visas på samma ställe som den var på innan man bytte skärmar. En skärm i tex mode blir upp till 9 block stor. En skärm i grafic-mode blir 33 block stor.

KOMMANDON FÖR HANTERING AV SKÄRMSIDOR

2–16 ssave

FUNKTION:	ssave sparar en skärmsida på disk
UPPBYGGNAD:	ssave du, "namn" (,n1)(,n2 to n3) där du = diskdrivens nummer (vanligen 8) n1,n2,n3 = tillhörande zoner
ANVÄNDNING:	den aktuella skärmsidan kan sparas. Zoner (decz) kan också sparas med hjälp av parametrarna n1 eller n2 to n3. Dessa parametrar är inte nödvändiga. Hela sidan kommer alltid att sparas.
EXEMPEL:	ssave 8,"prog1",1,12,30to70 ssave 8,"namn",0to127 Spar alla tillhörande zoner. Det genereras inga fel-meddelanden om man spar odefinierade zoner. ssave spar aktuell skärmsida (text eller grafisk), är den grafisk ger tillhörande zoner syntax error.

KOMMANDON FÖR HANTERING AV SKÄRMSIDOR

2–17 sload

FUNKTION:	sload laddar en skärmsida från disk.
UPPBYGGNAD:	sload du, "namn" där du = diskdrivens nummer (vanligen 8)
ANVÄNDNING:	skärmsidan visas direkt på skärmen.
EXEMPEL:	sload 8,"prg1"

KOMMANDON FÖR HANTERING AV SKÄRMSIDOR

2–18 sclear

FUNKTION: sclear raderar en skärmsida.

UPPBYGGNAD: sclear.

ANVÄNDNING: sclear raderar skärmen och data hämtad med hjälp av decz, reqz och inz men förstör ej definitionerna.

EXEMPEL: sclear

BESTÄM SKÄRMENS FÄRG

2–19 screen

FUNKTION: screen ändrar färg på skärm, ram och de tecken som visas av TOOL 64

UPPBYGGNAD: screen sc,br(,cr)
där sc =färgen på skärmen (0<=sc<=15)
br =färgen på ramen (0<=br<=15)
cr =färgen på tecknen (0<=cr<=15)(ej nödv.)

ANVÄNDNING: screen ändrar färg på skärm, ram och tecken som visas av TOOL 64.

EXEMPEL:

```
10 sclear:screen1,6,2:out"U.S.A",12,20:carget""
20 sclear:screen1,5,2:out"ITALY",12,20:carget""
30 sclear:screen0,2,7:out"GERMANY",11,7:carget""
40 sclear:screen1,2,2:out"JAPAN",11,9:carget""
```

ANTECKNINGAR: tabell för färgkod.

Svart	0
Vit	1
Röd	2
Turkos	3
Lila	4
Grön	5
Blå	6
Gul	7
Orange	8
Brun	9
Skär	10
Mörkgrå	11
Grå	12
Ljusgrön	13
Ljusblå	14
Ljusgrå	15

STOP och RESTORE återställer till orginalfärgerna med svart skärm och vita tecken.

3. HÖGUPPLÖSNINGSGRAFIK

3–1 Översikt

Följande avdelning kommer att handla om högupplösning. Högupplösningsskärmen har en upplösning på 320*200 punkter och är utformad som ett ritbord med origo i nedre vänstra hörnet. Varje punkt på skärmen definieras med en horisontell och en vertikal koordinat. Först går man över i grafik-mode (graphic) och använder de kommandon (move, draw, plot, point) som passar, därefter går man tillbaka till text-mode (text). Man kan även visa text (display) och ändra färger (color) i grafik-mode.

GRAFIKKOMMANDON

3–2 graphic

FUNKTION: graphic ställer om din 64 i grafik-mode

UPPBYGGNAD: graphic

ANVÄNDNING: graphic måste användas varje gång som grafiska kommandon skall utföras. Kommandot reserverar nödvändigt minnesutrymme för de grafiska kommandona.

EXEMPEL:

```
10 tline 40,1,1:tline 40,25,1
20 tcol 25,1,1:tcol 25,1,40
30 ssave 8,"RAM"
40 graphic
50 rem grafiska komandon (se följande sidor)
60 ssave 8,"GRAFIK"
70 rem vi är fortfarande i grafik-mod
```

ANTECKNINGAR: när ett skärmkommando ska användas (utom ssave, sload, sclear) måste man gå tillbaka till text-mode genom att använda text kommandot (sid 3–8).

GRAFIKKOMMANDON

3–3 move

FUNKTION: move flyttar cursorn till en angiven plats på skärmen.

UPPBYGGNAD: move x,y
där x =horisontell kordinat (0<=x<=319)
y =vertikal kordinat
(0<=y<=199)

ANVÄNDNING: move gör att du kan börja plotta punkter eller rita linjer på skärmen. Kommandot flyttar endast cursorn, ingenting visas på skärmen.

EXEMPEL:

```
10 graphic
20 move 0,0:rem flyttar till origo
30 i=160:j=100:move i,j
40 text
50 rem, fullständigare exempel följer
```

ANTECKNINGAR: origo i grafik-mode skiljer sig från origo i text-mode. Text läggs ut i skärmen uppifrån och ner varför origo är i övre vänstra hörnet i text-mode. I grafik-mode är det lämpligare att ha origo i nedre vänstra hörnet.

GRAFIKKOMMANDON

3–4 draw

FUNKTION: draw ritar eller suddar en linje från den punkt där cursorn befinner sig till den punkt som anges i kommandot.

UPPBYGGNAD: draw x,y,ty
där x =horisontal kordinat (0
y =vertikal kordinat (0
ty =numerisk variabel; 1 ritar
linje; 0 suddar linje

ANVÄNDNING: draw används varje gång man ska rita eller sudda en rak linje på skärmen.

EXEMPEL:

```
5 rem *** RITA EN 3-D KUB PÅ SKÄRMEN ***
10 graphic:sclear
20 a=0:b=20:c=40:d=100
25 move a,a:rem flyttar cursorn till origo
30 draw a,d,1:draw d,d,1:draw d,a,1:draw a,a,1
40 draw c,b,1:draw c+d,b,1:draw c+d,b+d,1
50 draw c,b+d,1:draw a,d,1
60 move d,a:draw d+c,b,1
65 move c,b:draw c,b+d,1
70 move d,d:draw d+c,d+b,1
80 carget""
90 text
```

ANTECKNINGAR: om du ersätter 1 med en 0 i draw kommandot så suddas kuben linje för linje. Carget utan variablerna l och c kan användas i grafik-mode.

GRAFIKKOMMANDON

3–5 plot

FUNKTION: plot ritar eller suddar en punkt.

UPPBYGGNAD: plot x,y,ty
där x =horisontell koordinat
y =vertikal koordinat
ty =numerisk variabel; 1 ritar
punkt; 0 suddar punkt

ANVÄNDNING: plot används när någonting annat än en rak linje ska ritas eller suddas på skärmen.

EXEMPEL:
```
5 rem*** RITA EN CIRKEL I EN KVADRAT ***
10 GRAPHIC:SCLEAR
20 a=0:d=100:e=50:pi=3.14159:pr=.01:dr=1
25 move a,a
30 draw a,d,dr:draw d,d,dr:draw d,a,dr:
draw a,a,dr
40 for i=0 to 2*pi step pr
50 plot e*(1+cos(i)),e*(1+sin(i)),dr
60 next i
80 carget ""
90 text
```

ANTECKNINGAR: om du kör samma program med dr=0 så suddas figuren.

GRAFIKKOMMANDON

3–6 point

FUNKTION: point testar om en punkt är ritad eller inte på skärmen.

UPPBYGGNAD: point x,y,ty
där x =horisontell koordinat
y =vertikal koordinat
ty =numerisk variabel som innehåller en 1:a om en punkt är
ritad annars en 0.

ANVÄNDNING: point kan användas för att kontrollera om den linje man tänker rita kommer att korsa en annan linje.

EXEMPEL:
```
10 graphic
20 a=0:d=100:dim a(100)
25 move a,a
30 draw a,d,1:draw d,d,1:draw d,a,1:draw a,a,1
40 for i=0 to 100
50 point 25,i,a(i)
60 next i
70 carget ""
80 text
90 FOR I = 0 TO 100 : IF A(I) <> 0 THEN PRINT "A(";I;")
100 NEXT I
```

GRAFIKKOMMANDON

3–7 display

FUNKTION: display skriver ut en sträng på skärmen i grafik-mode.

UPPBYGGNAD: display a$,l,c
där a$ =vilken sträng som helst
l =radnummer för startpunkt (1<=1<=25)
c =kolumnnummer för startpunkt (1<=c<=40)

ANVÄNDNING: display gör att du kan lägga ut text i grafik-mode.

EXEMPEL:

```
10 GRAPHIC:SCLEAR
20 e=50:pi=3,14159:pr=.01
30 for i=0 to 2*pi step pr
40 plot e*(1+cos(i)),e*(1+sin(i)),1
50 next i
60 display "DETTA ÄR EN CIRKEL",5,10
70 display "–––––––––",6,10
80 carget ""
90 text
```

GRAFIKKOMMANDON

3–8 text

FUNKTION: text ändrar från grafik-mode till text-mode.

UPPBYGGNAD: text

ANVÄNDNING: text är ett obligatoriskt kommando efter ett grafik-program för att behålla programmet i minnet.

EXEMPEL: se tidigare exempel.

ANTECKNINGAR: man startar i text-mode. Om man ger ett grafiskt kommando i textmode kommer det att utföras på den grafiska bildskärmen. Om man ger textkommando i grafik-mode så kommer detta att utföras utöver ssave och sclear kommandona.

GRAFIKKOMMANDON

3–9 color

FUNKTION: color lägger ut en färg-matris på 8×8 punkter i grafikminnet.

UPPBYGGNAD: color (–)l,c,lg,ln,co
där l =radnummer för startpunkt (1<=1<=25)
c =kolumnnummer för startpunkt (1<=c<=40)
lg =höjden
ln =längden

	co =ny färg (0 till 15 se sid 2–19) (–) =kommandot påverkar de punkter på skärmen som inte är tända. Skriver man ingenting på den positionen påverkar kommandot de tända punkterna.
ANVÄNDNING:	med color kan du bestämma att ett block på den grafiska skärmen skall ha en viss färg.
EXEMPEL:	10 graphic 20 fori=1to10 30 plot 125,100+i,1 40 draw 175,100+i,1 50 next i 60 carget" " 70 color –1,1,25,40,6 75 carget" " 80 color 1,1,25,40,7 85 carget" " 90 text

4. PROGRAMMERINGSHJÄLP

4–1 auto

FUNKTION:	auto numrerar raderna automatiskt.
UPPBYGGNAD:	auto n där n =uppräkningstal (1<=n<=255)
ANVÄNDNING:	för att börja den automatiska radnumreringen slår man in auto n och return, sedan skriver man in sin första programrad och slår return så kommer nästa radnummer att skrivas ut. Stängs av genom att man slår in auto utan parameter eller att man slår return direkt efter ett radnummer.
EXEMPEL:	auto 10 5 for i = 1 to 8 15 rem nästa programrad

PROGRAMMERINGSHJÄLP

4–2 delete

FUNKTION:	delete raderar programrader.
UPPBYGGNAD:	delete a (-b) där a och b =radnummer

ANVÄNDNING: delete raderar en eller flera programrader.

EXEMPEL: delete a
raderar rad a
delete a- raderar alla rader fr o m
rad a t o m sista raden.
delete -a raderar alla rader fr o m
första raden t o m rad a.
delete a-b raderar alla rader fr o m
rad a t o m rad b.

ANTECKNINGAR: delete kan användas både i direktmode och i program-mod.

PROGRAMMERINGSHJÄLP

4–3 renu

FUNKTION: renu omnumrerar raderna i ett program.

UPPBYGGNAD: renu (a(,b(,c))) (se användning)

ANVÄNDNING: renu omnumrerar alla raderna i ett program inklusive GOTO GOSUB. Det finns fyra varianter:
1. renu ändrar hela programet första raden blir 100 och uppräkningstalet är 10.
2. renu a ändrar hela programet första raden blir 100 och uppräkningstalet är a.
3. renu a,b ändrar hela programet första raden blir a och uppräkningstalet är b.
4. renu a,b,c ändrar del av programet med början från rad a den raden får nummer b och uppräkningstalet är c.

EXEMPEL: renu 55,100,10

PROGRAMMERINGSHJÄLP

4–4 dump

FUNKTION: dump listar alla variabler i ett program och deras värden.

UPPBYGGNAD: dump

ANVÄNDNING: kan användas när som helst i direktmod. Dump visar inte innehållet i area.

EXEMPEL: a=2:b=3:a$="TOOL 64"
dump
Resultat på skärmen
a=2
b=3
a$="TOOL 64"

PROGRAMMERINGSHJÄLP

4–5 error

FUNKTION: error visar var i en BASIC-rad felet finns.

UPPBYGGNAD: error
s. För att lokalisera felet slå in error. Då visas den felaktiga raden och felet kommer att visas reverserat.

EXEMPEL:
```
10 a$=str$(a
20 end
run
error
```

ANTECKNINGAR: inga andra kommandon får slås in före man slagit in error efter ett error-meddelande.

PROGRAMMERINGSHJÄLP

4–6 find

FUNKTION: find hittar en sträng eller ett tecken i ett program.

UPPBYGGNAD: find de ch de (a-b)
där de =ett tecken vilket som helst utom " och något av tecknen man letar efter.
ch =de tecken man letar efter.
a-b =kan man avgränsa området man letar i med, behöver ej användas.

ANVÄNDNING: letar reda på ett eller flera tecken i ett program.
EXEMPEL:
```
10 ?"GUL"
20 ?"KUL"
30 ?"MOGUL"
40 GUL=33
find/"GUL"/
find/GUL/
```
Prova båda möjligheterna och lägg märke till skillnaden.

PROGRAMMERINGSHJÄLP

4–7 trace

FUNKTION: trace utför programmet stegvis.

UPPBYGGNAD: trace

ANVÄNDNING: när man använder trace presenteras raden som utförs på skärmen. För att visa raden håller man ned shifttangenten, för att gå till nästa steg släpper man upp tangenten. För att stoppa trace i direkt-mode trycker man ned shift och run/stop samtidigt.

EXEMPEL:
```
10 ?"TOOL";
20 ?" 64"
trace
```

PROGRAMMERINGSHJÄLP

4–8 off

FUNKTION: off avslutar trace.

UPPBYGGNAD: off

ANVÄNDNING: avslutar trace i program-mode.

EXEMPEL:
```
100 rem program
110 trace
120 gosub1000
130 off
140 rem program
```

5. KOMPLETTERINGAR TILL BASIC

5–1 Översikt

I den här delen har vi samlat de kommandon som inte direkt kan förknippas med skärmen. Några kommandon som underlättar stränghanteringen (creatst, hunt). En förbättrad version av BASIC-kommandot (if then) som har ersatts med (if then else). Ett kommando som gör det möjligt att kopiera direkt från skärmen till en skrivare (hcopy). Två variabler är införda (joy(1),joy(2)) för att underlätta arbetet med joysticks.

KOMPLETTERINGAR TILL BASIC

5–2 hunt

FUNKTION: hunt talar om positionen på ett tecken i en sträng.

UPPBYGGNAD: hunt (–) chr$(ca),a$,pn
där ca =decimala värdet på ett ASCII-tecken
a$ =strängen som ska genomsökas
pn =talar om var i strängen sökningen skall börja
(–) =letar efter första tecken som skiljer sig från ca. Skriver man ingenting på den positionen letar den efter första tecken som är lika med ca.

ANVÄNDNING: att göra kontrolloperationer på strängar. Resultatet finns i zo.

EXEMPEL:
```
a$="abcdef"
hunt chr$(67),a$,1
print zo
3
hunt -chr$(67),a$,1
print zo
1
```

ANTECKNING: om inte karaktären finns är zo=0.

KOMPLETTERINGAR TILL BASIC

5–3 creatst

FUNKTION: creatst konstruerar en sträng.

UPPBYGGNAD: creatst a$,ln(,chr$(ca))
där a$ =namnet på strängen
ln =längden på strängen (l<=ln<=255)
ca =decimala värdet på ett ASCII-tecken

ANVÄNDNING: creatst konstruerar en sträng (a$) av tecken (chr$(ca)) med en bestämd längd (ln). Om man utelämnar (chr$(ca)) så fylls strängen med blanksteg.

EXEMPEL: creatst a$,80 konstruerar en sträng med 80 blanksteg creatst b$,10,"A" eller creatst b$,10,chr$(65) konstruerar en sträng med 10 stycken A.

KOMPLETTERINGAR TILL BASIC

5–4 if then else

FUNKTION: if then else har två möjliga kommandon beroende på vad villkoret ger för resultat.

UPPBYGGNAD:
```
10 if (villkor) then (kommando 1):
else (kommando 2)
eller
10 if (villkor) then (kommando 1)
20 else (kommando 2)
```

ANVÄNDNING: om villkoret är sant kommer kommando 1 att utföras och om villkoret är falskt kommer kommando 2 att utföras.
16 nästade if then else-kommandon är möjliga.

EXEMPEL
```
10 A=8:B=10
20 FOR I = 1 TO 15
30 IF I > A THEN PRINT"I>A ";: ELSE PRINT"I<=A
";
40 IF I > B THEN PRINT"I>B": ELSE PRINT"I<=B"
50 NEXT I
```

ANTECKNINGAR: kan bara användas i program-mode.

KOMPLETTERINGAR TILL BASIC

5–5 hcopy

FUNKTION: hcopy överför allting från skärmen till en printer. Om printern tillåter högupplösnings-grafik så kommer alla grafiska tecken på skärmen att skrivas ut på pappret.

UPPBYGGNAD: hcopy

ANVÄNDNING: överför dina grafiska program till papper.

EXEMPEL: 100 hcopy
Prova något av de tidigare exemplen.

ANTECKNINGAR: använder man detta kommando stoppas den interna klockan i din VIC 64 under utförandet.

KOMPLETTERINGAR TILL BASIC

5–6 joy(1);joy(2)

FUNKTION: joy(1);joy(2) är en variabel som håller reda på värdena

ANVÄNDNING: underlättar när man gör ett program för joysticks.

EXEMPEL:

```
10 GRAPHIC:SCLEAR
15 X=125:Y=199:AD=128:PLOTX,Y,1
20 CD=JOY(1):AD=CD
30 IF CD = 0 THEN 20
40 IF AD <128 THEN PLOT X,Y.O
50 IF CD> THEN CD=CD-128
60 ON CD GOSUB
110,120,100,140,150,160,100,180,190,200
70 PLOT X,Y,1 : GOTO20
100 RETURN
110 Y=Y+1:RETURN
120 Y=Y-1:RETURN
140 X=X-1:RETURN
150 X=X-1:Y=Y+1:RETURN
160 X=X-1:Y=Y-1:RETURN
180 X=X+1:RETURN
190 X=X+1:Y=Y+1:RETURN
150 X=X+1:Y=Y-1:RETURN
```

ANTECKNINGAR: joy(2) fungerar på samma sätt som joy(1).

6. DOS

Översikt

TOOL 64 underlättar arbetet för dem som använder diskettstation då den innehåller DOS-support som tillhandahåller snabbkommandon för arbetet med diskettstationen.

KOMMANDON:
> : ger dig disk error meddelande
>$: ger dig biblioteket på skivan
>I : initialiserar diskettstationen
>N:filnamn,idnummer
: formaterar en diskett
>R:nytt filname=gammalt filnamn

: byter namn på en fil
>S:filnamn
: suddar en fil
>C:nytt filnamn=gammalt filnamn
: kopierar en fil
>UI :reset DOS

Alla kommandon avslutas med att RETURN-tangenten trycks ned.

ANTECKNINGAR: utförligare beskrivningar finns i användar-manualen till din diskettstation.

A SUMMERING AV KOMMANDON

KOMMANDON

auto n	4– 2
carget a$(,l,c)	2–14
clear l,c,lg,ln	2– 4
clearz n	2–13
color (–)l,c,lg,ln,co	3– 9
creatst a$,ln(,chr$(ca))	5– 3
decz n,l,c,ln(,ty)(,f$)	2– 9
delete l1–l2	4– 2
display a$,l,c	3– 7
draw x,y,ty	3– 4
dump	4– 4
error	4– 5
find de sträng de	4– 6
graphic	3– 2
hcopy	5– 5
hunt (–),ca$,a$,pn	5– 2
if (v) then (k1) else (k2)	5– 4
inz n,a$	2–11
move x,y	3– 3
off	4– 8
out a$,l,c	2– 5
outz n,a$	2–12
plot x,y,ty	3– 5
point x,y,ty	3– 6
renu l1,l2,l3	4– 3
reqz n	2–10
rev l,c,lg,ln(,co)	2– 6
sclear	2–18
screen sc,br(,cr)	2–19
scroll l,c,ln,lg,ty	2– 7
sload 8,"namn"	2–17
ssave 8,"namn"(,n)(,n2–n3)	2–16

VARIABLER

B FELMEDDELANDE

BAD FORMAT:	printusingformatet stämmer inte överens med längdparametrarna i definitionen av decz (sid 2–9).
NO COLOUR:	en färgparameter saknas eller är det ett otillåtet värde på parametern (sid 2–19).
NO ZONE:	ett zon-kommando har utförts på en zon som inte finns (sid 2–9 tom 2–13).
OUT OF PAGE:	koordinaterna för origo, längd eller höjd är felaktiga



Box 42054, Vretensborgsvägen 8, 126 12 STOCKHOLM

KÅA Produkter AB